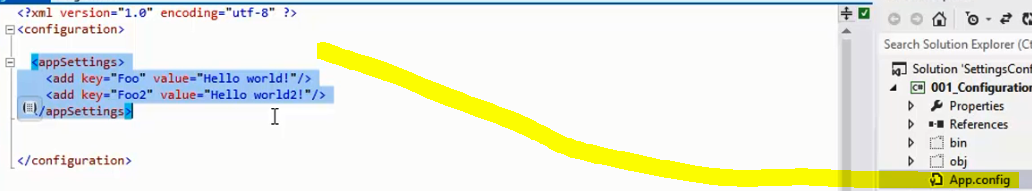


В App.config можно помещать любые элементы необходимые для работы приложения



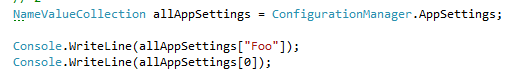
string value = ConfigurationSettings.AppSettings["Foo"]; - получить значение по индексатору

НО ДАННЫЙ КЛАСС УСТАРЕЛ!!!!

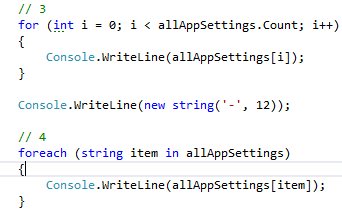


ConfigurationManager

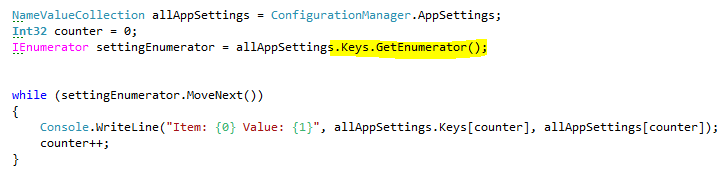
Получить коллекцию

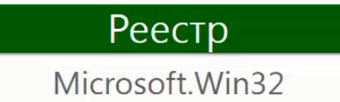


В коллекции получить элементы

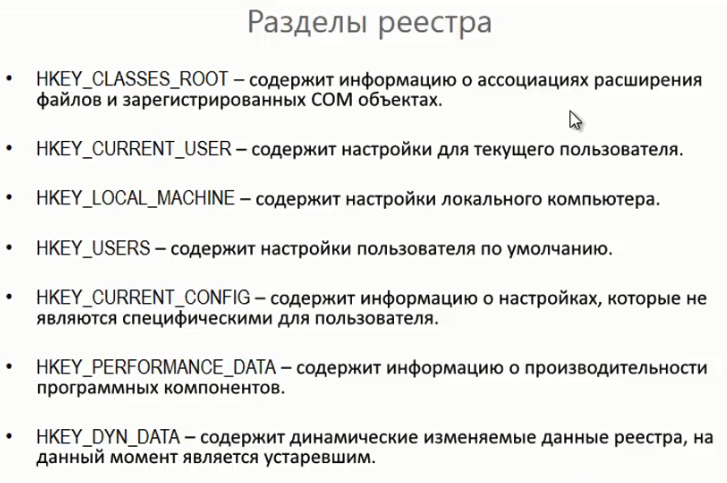


Или перебор элементов с помощью IEnumerator

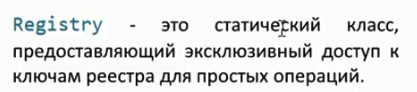






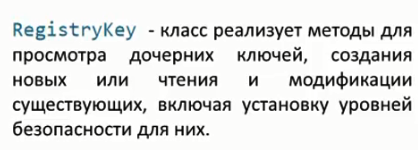


Registry

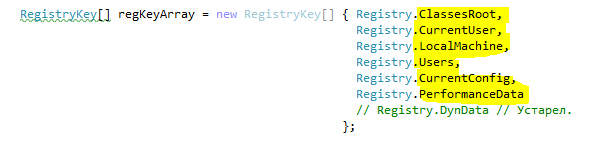
это класс, предоставляющий эксклюзивный доступ к ключам реестра для простых операций.

RegistryKey

класс реализует методы для просмотра дочерних ключей, создания новых или чтения и модификации существующих, включая установку уровней безопасности для них.

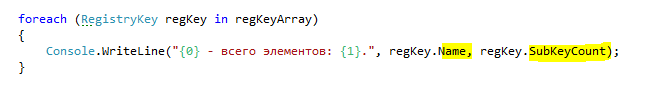


Создать массив для доступа к каждому разделу реестра



regKey.Name – имя реестра

regKey.SubKeyCount – количество ключей раздела



Процесс получения ссылки на объект RegistryKey называется открытием ключа.

Попытка открыть несуществующий ключ. Переменной будет присвоено значение NULL.

RegistryKey myKey = Registry.LocalMachine;

RegistryKey software = myKey.OpenSubKey("Software"); - открытием ключа

RegistryKey microsoft = software.OpenSubKey("Microsoft"); - открытием ключа

microsoft.Close(); - закрытие ключа

Создать собственный ключ в реестре

Для создания ключа в разделах отличных от Registry.CurrentUser нужны права администратора

Для этого нужно создать и изменить привилегии при запуске



// В этот раз открываем специально для записи.

RegistryKey myKey = Registry.CurrentUser;

// Второй аргумент (true) говорит о том, что будет совершаться запись.

RegistryKey wKey = myKey.OpenSubKey("Software", true);

// пытаемся создать новый ключ.

RegistryKey newKey = wKey.CreateSubKey("CyberBionicSystematics");

// Запись прошла успешно в HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node

// Закрыть ключ нужно обязательно.

myKey.Close();

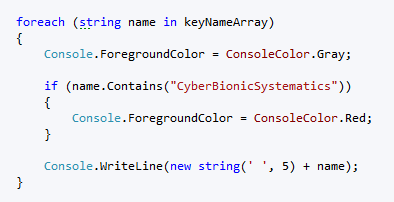
Удаление ключа из раздела регистра

RegistryKey myKey = Registry.CurrentUser;

// Для удаления тоже нужно иметь права редактирования.

RegistryKey wKey = myKey.OpenSubKey("Software", true);





wKey.DeleteSubKey("CyberBionicSystematics"); - удаление ключа, по имени

если ключа не существует будет выведена ошибка 

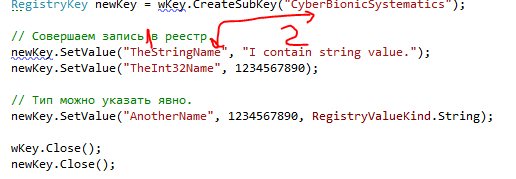
Запись в созданный ключ

newKey.SetValue("TheStringName", "I contain string value."); - 1 имя ключа который будет добавлен в раздел (CyberBionicSystematics) и 2й параметр значение

newKey.SetValue("TheInt32Name", 1234567890);

// Тип можно указать явно.

newKey.SetValue("AnotherName", 1234567890, RegistryValueKind.String);



Считывание, Получение данных с добавленных ключей

// Совершаем навигацию по реестру и открываем ключ для чтения,

// можно сразу указать весь путь, а не совершать навигацию поэтапно.

RegistryKey myKey = Registry.CurrentUser;

RegistryKey wKey = myKey.OpenSubKey(@"Software\CyberBionicSystematics");

// Читаем данные и конвертируем в нужный формат.

string Str = wKey.GetValue("TheStringName") as string; - передается ключ

int Int = Convert.ToInt32(wKey.GetValue("TheInt32Name"));

int Ant = Convert.ToInt32(wKey.GetValue("AnotherName"));

можно считать передав второй параметр (значение по умолчанию) если нет ЗНАЧЕНИЯ В КЛЮЧЕ с именем CurrText

textBox2.Text = (string)regKey.GetValue("CurrText", "NoValue");

УСТАНОВИТЬ ПРОГРАММУ В АВТОЗАГРУЗКУ

RegistryKey regKey = Registry.LocalMachine;

regKey = regKey.CreateSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run");

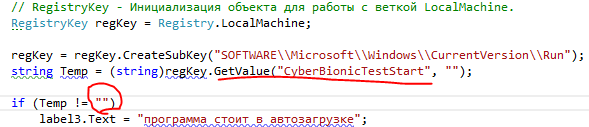
или открываем

regKey = regKey.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",true);

// Путь к данному приложению.

regKey.SetValue("CyberBionicTestStart", Application.StartupPath + [\\WindowsFormsApplication2.exe](file:///\\WindowsFormsApplication2.exe)");

проверить стоит ли программа в автозагрузки



Удалить программу с автозагрузки

